

**IN THE SPECIFICATION**

Please replace the numbered paragraph [0025], [0028], and [0029] with the following replacement paragraphs:

**[0025]** The PMR Quadtree 110 may be a data tree structure that is formed from the Z-values associated with the tiles 114A<sub>1</sub>-120D<sub>1</sub>. The PMR Quadtree 110 may be a layered data structure that includes four branches or leaves on each level to represent the Z-value of the tile and the associated sub-tiles. The PMR Quadtree 110 may include many different levels with each level having four leaves per node in the previous level. The PMR tile structure 108 may be represented by the base level 122, while the first level 124 represents the large tiles that include the grouping of four tiles, such as tiles 114A<sub>1</sub>, 114B<sub>1</sub>, 116A<sub>1</sub> and 116B<sub>1</sub> or tiles 114C<sub>1</sub>, 114D<sub>1</sub>, 116C<sub>1</sub> and 116D<sub>1</sub>, for example. The second level 126 may represent the individual tiles 114A<sub>1</sub>-120D<sub>1</sub> with the associated Z-values and spatial objects R-U. From the levels 122-126 of the PMR Quadtree 110, the spatial objects R-U and the associated Z-values of the tiles 114A<sub>1</sub>-120D<sub>1</sub> may be represented. For instance, the spatial object R may be represented by the Z-value of “03,” which is on the second level 126 and connected to the four node of the first level node, while the spatial object S may be represented by the Z-values of “00” and “01,” which are also on the second level 126 and connected to the first and second nodes, respectively. From the PMR Quadtree 110, the Z-values associated with the spatial objects R-U may be associated with entries 128-136 in the index table 90. For example, entry 134 represents a portion of the R-U object in index table 90.

**[0028]** FIG. 4 is a block diagram illustrating an exemplary flow of a query operation that may be implemented in embodiments of the present invention. The reference numeral

138 refers generally to the elements shown in FIG. 4. To perform a query operation on the spatial data in the base table 84 or the index table 90, the query operation may be divided into two modules, such as a filter module 140 and a refinement module 142. The filter module 140 may be utilized decompose the query window into Z-values and scan the index table 90 to provide results, such as a candidate table [[146]] 156 or results table 152, to the refinement module 142, which utilizes computational geometry to verify the results against the query window 144.

[0029] In the filter module 140, the query window 144 is decomposed into non-intersecting Z-values and referenced against the index table 90. The query window 144 may describe a spatial area of interest (e.g., a circle, square, or other polygon), which may include roads, streets, cities, and/or other geographical locations. The spatial area may encompass the spatial objects that intersect with the spatial area or are within the spatial area defined by the query window 144. The query window 144 is decomposed at the decompose window operator 98 into a window table 148 that contains a list of Z-values that represent the query window 144. The window table 148 is provided to a join operator, such as a nested-join relational operator 100. The scan index operator 96 scans the index table 90 to form a scan index table 150 that is also provided to the nested-join relational operator 100. The nested-join relational operator 100 may provide the Z-values or may provide a scan range for each of the Z-value in the window table 148 to the scan index operator 96. The nested-join relational operator 100 may perform operations to determine pairs of Z-value equivalent entries from the window table 148 and the scan index table 150. From the operations, the nested-join relational operator 100 produces a result, such as entries in the results table 152 or a list of OIDs. The

result includes candidate objects that may satisfy the spatial relationship of the query window 144. The results table 152 may be provided to a GroupBy operator 102 or directly to the join operator 155. The GroupBy operator 102 may create a candidate table [[146]] 156 by removing duplicate entries from the results table 152. The GroupBy operator 102 may provide candidate table [[146]] 156 to the join operator 155 in the refinement module 142.